Chapter 2: Access Protocols

1. Review of the Poisson Process and the Exponential distribution

The Poisson process will be used to model the arrivals of messages in a communications system. It is an accurate model of a system with a large number of users who generate messages independently of one another, and independent of when they generated their last message.

With a Poisson arrival process, the interval between arrivals is an exponential distribution.

1.1 The Poisson Arrival Process

The probability of n arrivals in any interval τ is:

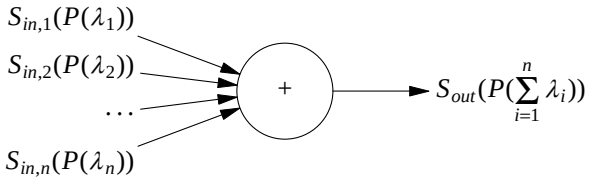
$$P_n(\tau) = e^{-\lambda\tau}\frac{(\lambda\tau)^n}{n!}, \text{ for n=0,1,...}$$

1.2 The Exponential Interarrival Time Distribution

- t_n is the time of the n^{th} arrival, and
- The time between the $(n-1)^{th}$ and the n^{th} arrival is $\tau_n = t_n - t_{n-1}$
- The probability that $\tau_n \geq t$ is the probability of no arrivals in the interval t:
  $P(\tau_n \leq t) = 1 - P_0(t) = 1 - e^{-\lambda t}$, for $t \geq 0$,
  The probability distribution is independent of n, the message number.
- The interarrival times between messages are exponentially distributed:
  $p(\tau) = \lambda e^{-\lambda\tau}$

1.3 Properties

1. $E(\tau) = 1/\lambda$, $Var(\tau) = 1/\lambda^2$
   The arrival rate is λ messages/second
2. The exponential distribution is memoryless:
   $$P(\tau > t + t_1 \,/\, \tau > t_1) = \frac{P(\tau > t + t_1)}{P(\tau > t_1)} = \frac{e^{-\lambda(t+t_1)}}{e^{-\lambda t_1}} = e^{-\lambda t} = P(\tau > t)$$
   The time until the next arrival is independent of how long the system has been waiting.
3. The sum of Poisson processes is a Poisson process with the output rate equal to the sum of the input rates



Warning: The Poisson arrival process has been mistakenly applied to ATM networks in which messages are partitioned into many cells that arrive one after another. *The arrivals are not independent of one another.* As a result, the buffer sizes in the first generation of ATM switches were smaller than needed.

2. Aloha Protocols

2.1 Protocols

Reference 1 section 4.2

The sources share a single channel, but are physically separate.

- The packets have a fixed length.
- Packet arrivals at the sources are independently, and form a Poisson arrival process with an arrival rate of *S* packets per packet transmission time. *(Note that the arrival rate is normallized to the packet transmission time. This allows us to analyze a network without considering the transmission rate of the channel.)*
- If the transmission from two or more sources overlap at the receiver, the transmissions collide, and the packets must be retransmitted.
- A source picks the time that it waits from an exponential distribution, so that the retransmitted messages also form a Posison process.
- The arrival rate of messages from the source and retransmitted messages is a Poisson process with arrival rate *G*.
- The exponential retry distribution can wait a very long time before retransmitting. In most systems that have been implemented, the retry time is picked from a uniform distribution, which has an upper bound on the retry time. The exponential distribution is an approximation that is used in most analyses.

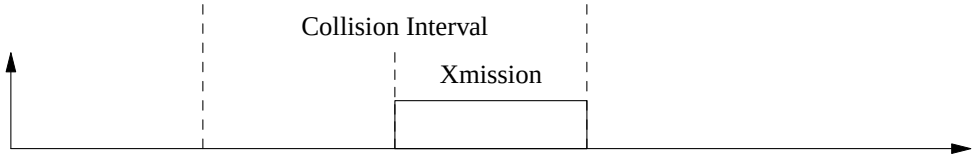
There are two types of Aloha systems, slotted and unslotted.

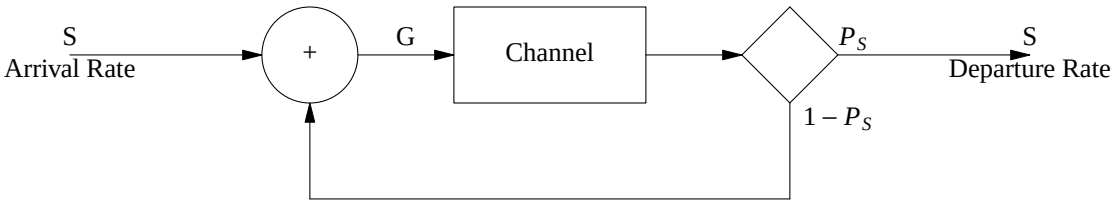
2.1.1 Unslotted Aloha

In Unslotted Aloha, the transmissions from the sources are not coordinated, and sources can transmit whenever they like.
Unslotted Aloha was first used by Norm Abramson at the University of Hawaii in 1970 to share a radio channel between computer terminals on the islands and a main computer. [2]

A collision occurs if:

1. A packet starts transmitting before the current packet arrives, and is still transmitting when it arrives, or
2. A packet arrives while the current packet is transmitting





- The arrival Process is Poisson with S arrivals/packet transmission time
- The transmission rate, G, is also Poisson with rate $G = S + (1 - P_S)G$, so that $S = P_S G$
- The departure rate is the rate at which packets are successfully transmitted, and equals S when all the packets that arrive eventually get through.
- P_S is the probability of a successful transmission = the probability of no other arrivals in the collision interval $= e^{-2G}$
- Departure rate = Ge^{-2G}
- The maximum departure rate occurs when:

$$\frac{d}{dG}\left(Ge^{-2G}\right) = e^{-2G} - 2Ge^{-2G} = 0$$

$G = 1/2$

- The maximum departure rate = $\frac{1}{2}e^{-1} \approx 0.184$

2.1.2 Slotted Aloha

In Slotted Aloha the transmissions from a source are coordinated so that colliding packets overlap entirely.

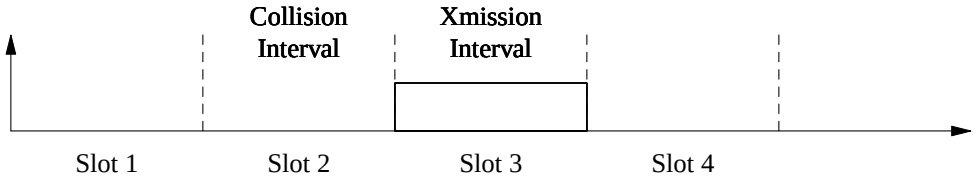
Slotted Systems:

A. Small distances
   - — Multiple sources and destinations
   - — Slighty longer slots than required for transmission
   - — Transmissions between different sources in the area arrive at all of the receiver within the slot time

B. Single receiver
   - — Base station in cellular radio
   - — Transmitters are synchronized to arrive at the receiver at the same time.

C. Single regenerator
   - — Satellite or head-end in a CATV network
   - — Multiple sources and destinations
   - — All sources transmit on one freqeuncy to the regenerator.
   - — Transmission are synchronized to arrive at the regenerator at the same time
   - — The regenerator receives the signals and retransmits on another frequency to the receivers

Collision Interval
All packets that arrive during slot 2 are transmitted at the beginning of slot 3
If 2 or more packets arrive during slot 2 there is a collision in slot 3



Departure rate = Prob of 1 transmission in a slot = Ge^{-G}

- Maximum departure rate occurs at:

  $$\frac{d}{dG}\left(Ge^{-G}\right) = e^{-G} - Ge^{-G} = 0$$

  $G = 1$

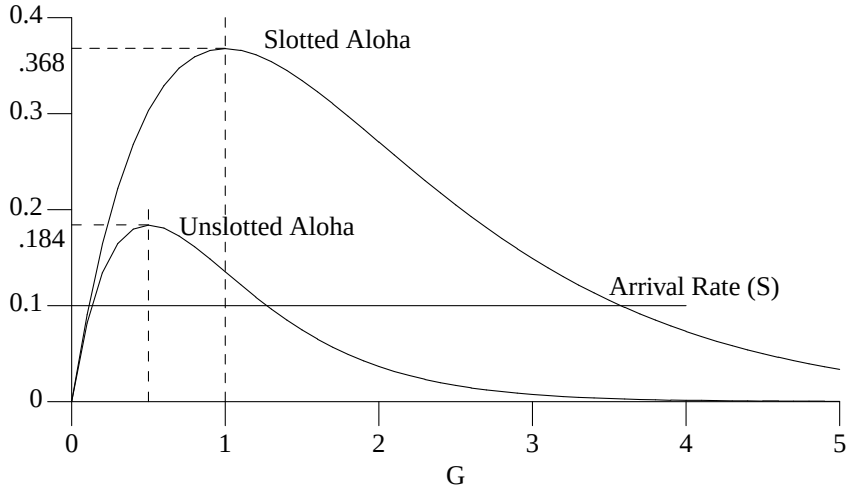
- Maximum departure rate = $e^{-1} \approx 0.368$
- At the maximum departure rate:

  Pr(empty slot) = $e^{-G} = e^{-1} \approx .368$

  Pr(successful xmission) = $Ge^{-G} = e^{-1} \approx .368$

  Pr(collision) = $1 - e^{-G} - Ge^{-G} \approx 1 - 2*.368 = .264$

2.2 Comparison of unslotted and slotted Aloha



In equilibrium:

The arrival rate = departure rate

$S = Ge^{-2G}$ for unslotted Aloha and $S = Ge^{-G}$ for slotted Aloha

no equilibrium point for S > Maximum departure rate

2 equilibrium points for S < Maximum departure rate

The left most point is a *stable* equilibrium point, while the right most point is an *unstable* equilibrium point.
This will become evident after a more detailed analysis

Average Delay:

— The average number of times a source transmits is $\frac{G}{S}$.

— The average number of times a source retrys, when there is an equilibrium point, is $\bar{R} = \left(\frac{G}{S} - 1\right)$

$\bar{R}_{unslotted} = e^{2G} - 1$
$\bar{R}_{slotted} = e^{G} - 1$
At $G = G_{\max}$, $\bar{R} = e - 1$, for both systems

— If the average time between retrys is $\bar{W}$, the average delay is $T = 1 + \bar{R}(1 + \bar{W})$

Unslotted Aloha is simpler to implement than slotted Aloha, - no slot timing

Unslotted Aloha can operate with variable size packets

2.3 Capture effect in radio - perfect capture [3]

So far we have assumed that we only succeed if one station transmits in a slot.
This is an accurate model for local area networks where the transmission distances are small.
In radio networks the power at the receiver is proportional to $\frac{1}{d^i}$, where d is the distance from the source to the receiver and $i = 2, 3$ *or* 4, depending on the type of antennea that is used.
If all sources transmit with the same power, and one source has a much higher power at the receiver, the signal from that source is correctly received, even though there is a collision.

If you are listening to the radio and receive both a nearby station and a distant station, the distant station may appear as back ground noise, while you can still clearly hear the nearby station.

In a digital system, where we make a decision that a 1 or a 0 was transmitted, the distant station appears a noise that increases the probability of error, but the error rate may still be low.
In perfect capture we assume that when multiple stations transmit simultaneously, the power from one of the stations is much higher, and that station wins when they collide.

Perfect capture is an approximation that is used to determine the maximum throughput of radio network.

2.4 A more exact model of slotted Aloha

2.4.1 Backlogged Sources

— The probability of a retry packet arriving is a function of the number of sources that are retrying.
The sources that are retrying are referred to as backlogged sources.

— The time until a backlogged source retransmits is a negative exponential distribution, with an arrival rate λ_r.
A backlogged source may only retransmit in a slot once, and the probability that it retransmits is $q_r = 1 - e^{-\lambda_r}$.
The probability that a backlogged source retransmits in a slot is independent of how many slots it has waited.
The probability that a source transmits in the i^{th} slot after it is backlogged is a geometric distribution $p(i) = q_r(1 - q_r)^{i-1}$
The average value of i is $\frac{1}{q_r}$, so that, the arrival rate for a backlogged source is q_r transmissions per slot.

— If there are n backlogged sources, the prob that i backlogged sources retransmit in a slot is the binomial distribution
$Q_r(i, n) = \binom{n}{i}(1 - q_r)^{n-i} q_r^i$.
The average number of transmissions in a slot is nq_r
Therefore, the arrival rate of retransmitted packets, with n backlogged sources is $\lambda_r(n) = nq_r$ packets per slot.

2.4.2 Finite Source Model

If there are a finite number of sources, m, the probability of a new packet arriving in a slot decreases when there are backlogged sources.

It is assumed that a new message does not arrive at a source while it is backlogged. There is no buffering at the sources.

— The time between arrivals from a source that isn't backlogged is a negative exponential distribution with arrival rate S/m.

— A source may only transmit once in a slot, and the probability that it transmits is $q_a = 1 - e^{-S/m}$
As before, the arrival rate for sources that are not backlogged is q_a transmissions per slot.

— The probability of i new arrivals when n of the m users are backlogged is $Q_a(i,n) = \binom{m-n}{i}(1-q_a)^{m-n-i}q_a^i$
The arrival rate for sources that are not backlogged is $\lambda_a(n) = (m-n)q_a$.

— The total arrival rate of new and backlogged source with a backlog of n is $\lambda_T(n) = \lambda_r(n) + \lambda_a(n) = nq_r + (m-n)q_a$.

— The departure rate, λ_D, is the probability of successful transmission, the probability that one new source or backlogged source transmits in a slot. It is the throughput of the system.

$$\lambda_D(n) = Q_a(1,n)Q_r(0,n) + Q_a(0,n)Q_r(1,n)$$

$$= (m-n)q_a(1-q_a)^{m-n-1}(1-q_r)^n + nq_r(1-q_r)^{n-1}(1-q_a)^{m-n}$$

$$= \left((m-n)\frac{q_a}{1-q_a} + n\frac{q_r}{1-q_r}\right)((1-q_a)^{m-n} + (1-q_r)^n)$$

In order to get the departure rate in the form of the simpler model, we will assume that there are a large number of sources, and that the arrival rate from individual sources is small.
$q_a \ll 1$ and $q_r \ll 1$.
We use the approximation, $(1-x)^y \approx e^{-xy}$, for small x, so that.

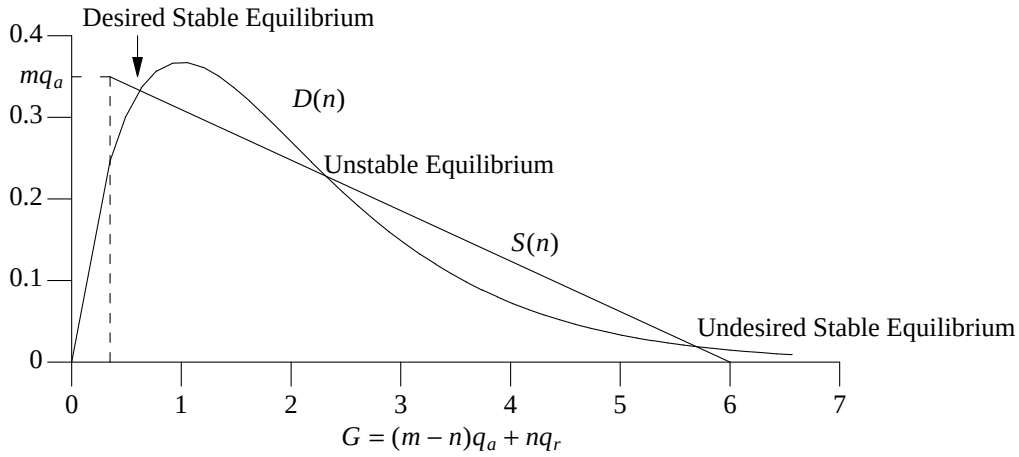
$$\lambda_D(n) \approx \left((m-n)\frac{q_a}{1-q_a} + n\frac{q_r}{1-q_r}\right)e^{-((m-n)q_a+nq_r)}$$
$$\approx ((m-n)q_a + nq_r)e^{-((m-n)q_a+nq_r)}$$

$$= G(n)e^{-G(n)}$$

where, $G(n) = (m-n)q_a + nq_r = \lambda_T(n)$

In the figure, the number of users $m = 40$, $S < .35$, so that $q_a < .00875$, and $q_r = .15$.

- Arrival Rate = $S(n) = (m-n)q_a$
- $G(n) = (m-n)q_a + nq_r$
- Departure rate = $G(n)e^{-G(n)}$



This system has 3 equilibrium points, where the arrival rate equals the departure rate.

Because $q_a < q_r$.
When the number of backlogged users, *n*, increases, G also increases
When *n* decreases, G also decreases

At the **stable** equilibrium points, when *n* changes, which moves the system away from the stable point, the departure rate also changes in the direction forces the system back to the equilibrium point.

— If *n* increases, G increases,
The departure rate is greater than the arrival rate,
Therefore, *n* decreases and the system moves back to the equilibrium point.

— If *n* decreases, G decreases,
The departure rate is less than the arrival rate
Therefore, *n* increases and the system moves back to the equilibrium point

At the **unstable** equilibrium point, when *n* changes, which move the system away from the equilibium point, the departure rate changes in a direction that forces the system to move futher away from the equilibrium point.

— If *n* increases, G increases,
The departure rate is less than the arrival rate
Therefore, *n* increases more, and the system moves further from the equilibrium point

— If *n* decreases, G decreases,
The departure rate is higher than the arrival rate
Therefore, *n* decreases more, and the system moves further from the equilibrium point.

When the system is in the unstable equilibrium point, it will move to the left stable equilibrium point if one of the backlogged users sucessfully transmits, and will move to the right stable equilibrium point if a new source collides.

The left most stable equilibrium point is the desired operating point because it has a higher throughput than the right most point.
Since the *G* of the right most stable equilibrium point is larger, the number of retrys is higher and the delay is bigger.
The right most stable equilibrium is a condition in which most of the sources are backlogged and are

retrying.

Adaptive Retry Intervals The arrival rate of a system is not constant, but changes continuously.
When the arriving load is light, retrying quickly will reduce the average delay.
However, when the arrival rate is high, retrying quickly will overload the system and result in excessive delays

We have seen that there may be more than one stable operating point in a system.
When the system is in an undesirable stable operating point we can adjust the retry interval to force the system to move to the desired stable operating point.

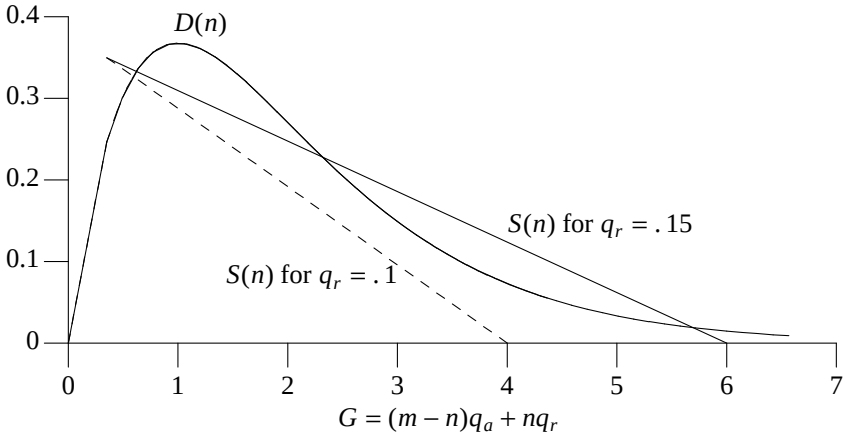
A system with three equilibrium points is defined to be a moderatly loaded system.
Before discussing the means for adjusting the retry interval, we will look at two other types of systems, lightly loaded sytems and heavily loaded systems.

1. Lightly loaded system

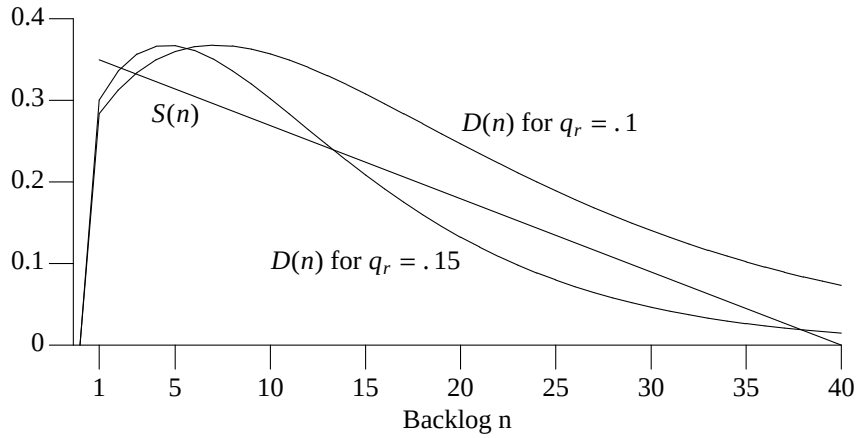
A lightly loaded is a system with one stable equilibrium point, at the desired stable equilibium point.

If we have a moderately loaded system, that is operating at the undesirable, stable operating point, we can transform the system into a lightly loaded system by increasing the time between retrys.
This forces the system to return to the desirable stable operating point.

In the figure we decrease q_r from .15 to .1



In the previous figure, we plot the departure rate as a function of G. As a result, the curve for the departure rate is the same for all values of q_r and there is a different curve for the arrival rate for each value of q_r.
It is instructive to plot the departure rate as a function of the backlog, n. This results in a different curve for the departure rate for each value of q_r, but a single curve for the arrival rate.



We see that when we decrease q_r at the undesirable equilibrium point, the departure rate increases above the arrival rate. n will decrease and the curve moves toward the desired equilibrium point.

Once we reach the stable equilibrium point, we should increase q_r
Although the throughput for the 2 desirable stable equilibrium point is about the same, the one with q_r = . 15 is preferred.
The larger q_r decreases the average retry interval, and results in a smaller average delay

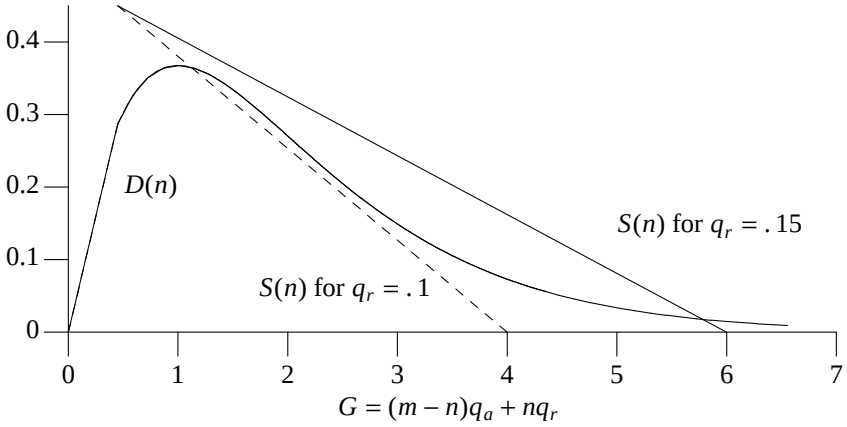
A source can use its value of n to determine which equilibrium point the system is in.
When n > 20, a source can adjust its q_r to move toward the desirable equilibrium.

2. Heavily Loaded System

A heavily loaded is a system with 1 stable operating point in which most of the users are backlogged

In the figure S is increased from .35 to .45
Since .35 is below the maximum throughput .368, this system can never be heavily loaded

In this figure we reduce the load by reducing q_r.
In very heavily loaded systems, decreasing q_r may increase the average delay above an acceptible amount.
We can also reduce the load by flow controlling the arrival rate S.



2.4.3 Infinite Source Model

In some systems, there are a large number of possible sources, with very small arrival rates, relative to the number of sources that are actively trying to transmit at any particular time, $m \gg n$, and we assume that the number of sources is infinite.

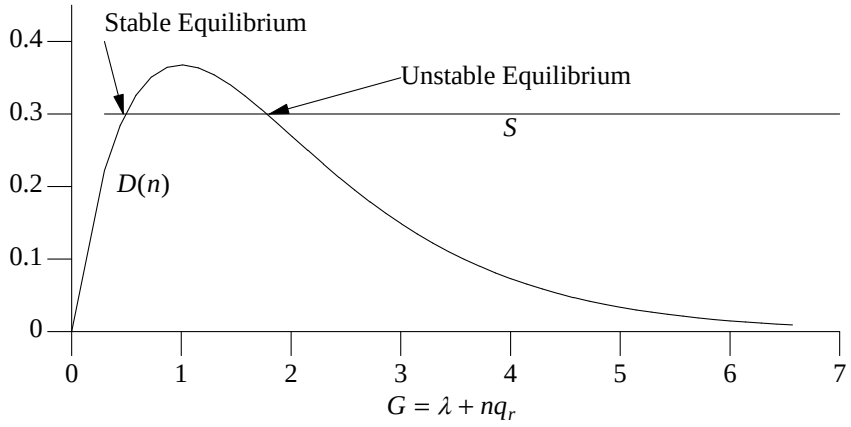
In the infinite source model we assume that S is not a function of n. $S(n) = S$.

The probability of j new arrivals in a slot is $P_j(S) = \frac{S^j}{j!} e^{-S}$

The probability of j retrys given n is the same as in the finite source model.

As a result, $D(n) = G(n)e^{-G(n)}$, where $G(n) = S + nq_r$

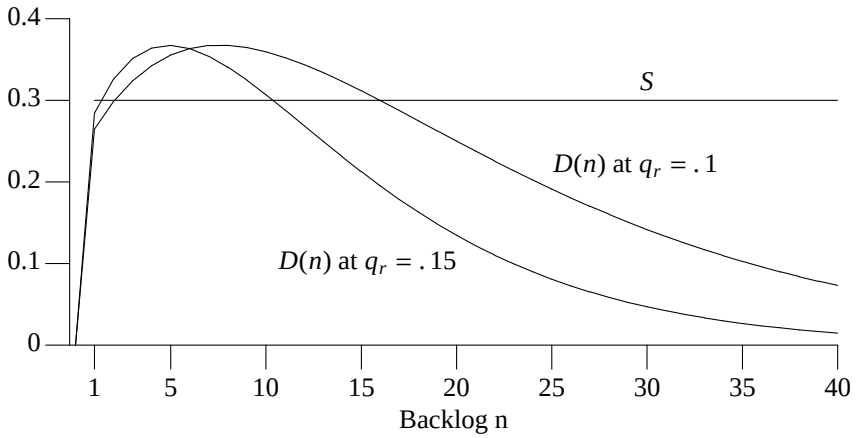
In the figure we reduce S to .3, to make the equilibrium points easier to see.



Once the stystem drifts to the right of the unstable equilibrium point, the backlog, and the delay, becomes infinite

This instability demonstrates the need for an adaptive technique

Since the arrival curve doesn't change as we change q_r, plotting with respect to G doesn't demonstrate the effect of changing q_r.



Decreasing q_r increases the backlog at the unstable equilibrium point and deceases the drift, the rate at which the backlog grows, to the right of that point

When the backlog starts growing in an unbounded way, we must decrease q_r, so that the departure rate is greater than the arrival rate for the current value of n.

How should we adapt q_r

1. Maximize throughput

The maximimum throughput, .368, occurs at $G(n) = 1$

— If we knew m, n, and q_a we could pick q_r so that

$$G(n) = (m - n)q_a + nq_r = 1$$

$$q_r = \frac{1 - (m - n)q_a}{n}$$

— It is very difficult, if not impossible, to calculate the optimum value of q_r.

The number of sources, m, may be known.

The average arrival rate per source per slot, q_a, is assumed to change continuously, but the short term average can be estimated as the average number of successes per slot divided by m. (This assumes that the departure rate = arrival rate).

However, the backlog n changes from slot to slot. It is not usually possible to determine the current value from the past history of the system because we cannot determine the number of stations that have collided or know how many of those stations are new arrivals.

— We can construct an adaptive algorithm that lets q_r approach the optimum value, without knowing m, n, or q_a.

At the optimum value of q_r, $G = 1$.

When $G = 1$, $P_{collision} = .264$

We can count the collision slots and estimate the $P_{collision}$.

We are interested in a very short term average, since n is changing from slot to slot, so this approach is only approximate.

If $P_{collision} < .264$ increase q_r

If $P_{collision} > .264$ decrease q_r

2. Ethernet: binary exponential back-off

Each source changes q_r independently, dependent on its recent failure history.

Instead of q_r being used in a geometric distribution, with a long tail, it is used as the probability of access in a uniform distribution, with an upper bound on the retry delay.

$q_r = \left(\frac{1}{2}\right)^{n_r}$, where n_r is the number of times that the source has tried to transmit the current packet.

first retry, one attempt has been made to transmit the packet, q_r = 1/2, transmit with probability 1/2 in each of next 2 slots

second retry, q_r = 1/4, transmit with probability 1/4 each of next 4 slots

...

This technique has worked well because:

n is usually small, and terminals succeed quickly

The system seeks the desirable stable operating point because heavily used systems have fewer retrys

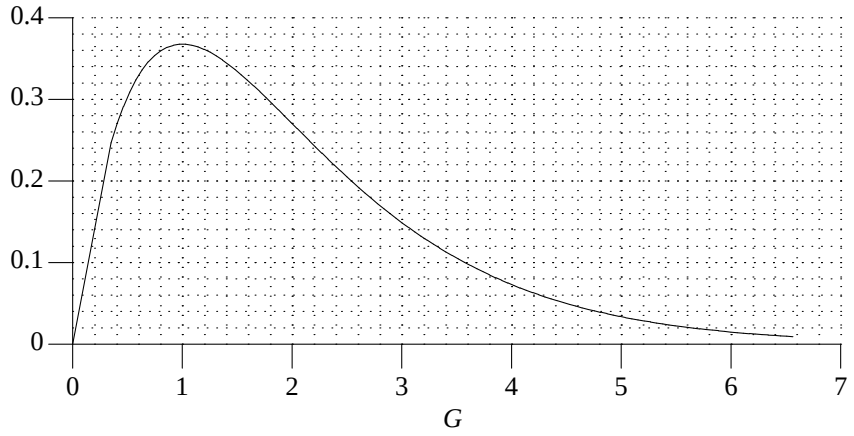
The technique is unfair because sources that have been backlogged wait longer than sources that have just failed.

Such is life

Homework

The departure curve for slotted Aloha $D = Ge^{-G}$ is plotted. For a system with m=100 users

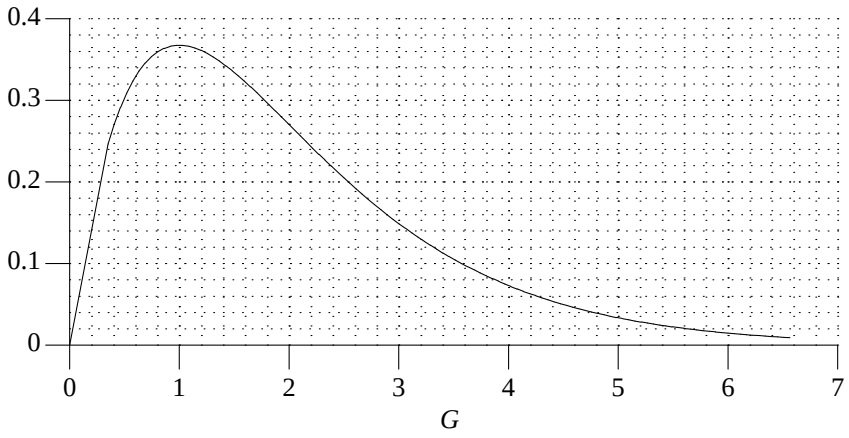
A. Draw the arrival line with an arrival rate q_a = .0035 arrivals per slot, and a retry rate q_r = .06 retrys per slot.
What is the x,y cordinates of this line when the number of backlogged users, n=0 and when n=100?
Label the desired stable equilibrium point, the undesired stable equilibrium point, and the unstable equilibrium point.
Determine the (x,y) coordinates of these points from the graph.



B. What is the throughput, average number of retrys and the average delay at the desired and undesired stable equilirium points?

When we are in the undesired stable equilibrium point, we lower q_r until the system has 1 stable equilibrium point.

C. Draw the arrival line for the largest q_r that has a single stable equilibrium point.



D. Determine q_r, and the throughput at the stable operating point from the curve.
Calculate the average number of retrys, and the average delay at the stable operating point.

3. Tree Splitting

3.1 Splitting Algorithms

Splitting Algoritms: are an alternative method to resolve contention between sources that have collided while trying to access a communications channel. Reference [1], section 4.3

Our first strategy to resolve contention was to have backlogged sources retry after a randomly selected interval.
When a backlogged source retransmits it is contending with other backlogged sources as well as newly arrived sources.
As long as the probability of success after each retry is greater than zero, the backlogged source eventually succeeds.
We noted that this strategy may have more than one stable operating point. In a desirable stable operating point there are few backlogged sources on the average and average number of retrys and the time it takes to resolve contention is relatively small. However, when there are many backlogged sources, the number of retrys may be large and the time it takes to resolve collisions is also large.

The splitting algorithms use a different strategy to resolve contention.

A. Once a collision occurs we enter a contention resolution phase.
Any new sources that arrive during a contention resolution phase wait until the contention is resolved before they transmit.

B. During the contention resolution phase we split the set of backlogged sources into smaller groups, or subsets.
We will describe three techniques for partitioning the backlogged sources into subset:

1. Using fixed numbers, or addresses, that are assigned to each source.

2. Randomly selecting a subset for each backlogged source.

3. And, based on the arrival time of the packet at the source.
In this case we can guarantee that the first backlogged source that arrives is the first to successfully transmit.

C. In the first slot following a collision, all of the sources in the first subset transmit.
If there aren't any backlogged sources in the subset, the slot is empty.
If there is one backlogged source in the subset, it successfully transmits.
In either of these cases, there are no backlogged sources in the first subset and the sources in the second subset transmit in the next slot.

D. If there is more than one backlogged source in the first subset, there is another collision.
We divide the sources in the first subset into smaller subsets, and resolve the contention between these sources before the sources in the second subset transmit.
We continue to divide subsets into smaller and smaller subsets until eventually there is 0 or 1 sources in a subset.

E. The same algorithm for dividing sources into subsets and deciding when to transmit is used at each of the backlogged or waiting sources.
The algorithm is completely distributed.
At each step a source knows which subset it is in, but doesn't know which sources are in the other subsets.

F. Splitting algorithms resolve contention between backlogged sources more quickly than random retrys, particularly when there are a large number of backlogged sources.
Splitting algorithms are especially useful in sensor networks where an external stimulous causes a large number of sources to transmit at about the same time.

3.1.1 Splitting Based on the Terminal Address

When a source is involved in the first collision, the high order bit of its address is used to resolve the collsion.

When a source is involved in a second collision, the next highest order bit of its address is used to resolve the collision,

. . .

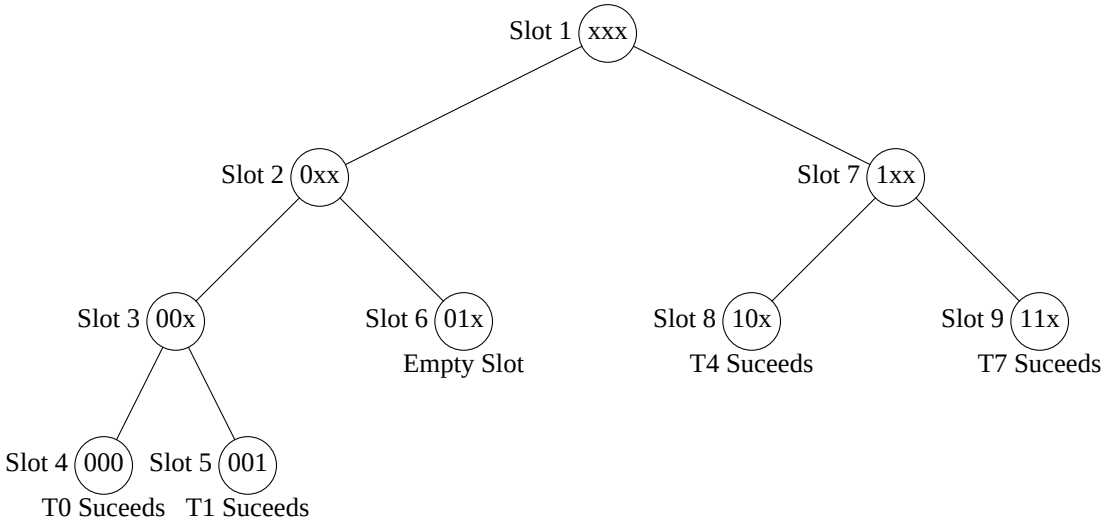
Example: Following a contention resolution interval, sources 0,1,4,7, with addresses 000, 001, 100, 111, are waiting to transmit and collide in the first slot.

Operation of the splitting rule

At each step, the permission rule indicates the addresses of the sources that can transmit. "X" is don't care, and indicates that the address bit can be either 0 or 1.

Slot	Permission	Transmitters	Result
1	XXX	000, 001, 100, 111	Collision
2	0XX	000, 001	Collision
3	00X	000, 001	Collision
4	000	000	000 Succeeds
5	001	001	001 Succeeds
6	01X	none	Empty
7	1XX	100, 111	Collision
8	10X	100	100 Succeeds
9	11X	111	111 Succeeds
10	XXX	New Contention	

Representation of the Transmission Pattern as a Tree



Implementation:

In concept, each source maintains a push-down, pop-up stack of the transmission privileges in the next slot.

— Initially, at the beginning of a contention resolution interval, the stack contains $(X_1, X_2, \ldots, X_N)$ and all sources have permission to transmit.

— At the beginning of a slot

- The top entry is popped off the stack
- If the entry gives a source permission to transmit, it does.
- If the slot is empty, or if there is a successful transmission, nothing is placed on the stack
- If there is a collision, and the entry that was popped from the stack is
  $(b_1, \cdots, b_i, X_{i+1}, \cdots, X_N)$,
  the high order don't care bit is split into two entries with 1 and 0,
  $(b_1, \cdots, b_i, 1, X_{i+2,} \cdots, X_N)$ and $(b_1, \cdots, b_i, 0, X_{i+2,} \cdots, X_N)$
  that are pushed back onto the stack.

— When the stack is empty at the end of a slot, $(X_1, X_2, \ldots, X_N)$ is pushed onto the stack, and the next collision resolution interval starts.

In the example, the stack that is maintained at each source is:

Step	1	2	3	4	5	6	7	8	9	10
Stack	XXX	0XX 1XX	00X 01X 1XX	000 001 01X 1XX	001 01X 1XX	01X 1XX	1XX	10X 11X	11X	XXX

The complete stack contains more information than a source needs.

At any step there is at most one stack entry that gives a source permission to transmit.

A source can determine when it can transmit by maintaining 3 pointers.

1. C_B: The number of stack entries that precede the entry that give it permission to transmit. This is the number of collisions that must be resolved before a source transmits.
2. C_T: The total number of entries in the stack. This is used to determine when the stack is empty and the next collision interval starts. It is the total number of collisions that remain in the contention interval.
3. N_B: The next high order bit at the source. When a source collides, this bit determines if its next transmission is pushed onto the stack first or second.

Algorithm implemented at each source:

— At the beginning of a contention interval

$C_T = 1$,

$$C_B = \begin{cases} 0 & \text{if } \textit{the source has a packet transmit} \\ 1 & \text{if } \textit{the source does not have a packet transmit} \end{cases}$$

$N_B = 1$

— At the beginning of a slot

- If $C_B \neq 0$

  If a collision occurs, increment C_T and C_B

  If there is a successful xmission, or the channel is idle, decrement C_T and C_B

- If $C_B = 0$

  Transmit a packet

  If the xmission is successful:

  Decrement C_T

  $C_B = C_T$

  If there is a collision

  increment C_T

  C_B = value of N_B^{th} bit of the address, 0 or 1.

  Increment N_B

— When $C_T = 0$ start a new contention interval.

Example: Counters at source 4 - 100

Slot	C_T	C_B	N_B	Action	System
1	1	0	1	Xmit	collision
2	2	1	2	Wait	collision
3	3	2	2	Wait	collision
4	4	3	2	Wait	success
5	3	2	2	Wait	success
6	2	1	2	Wait	empty
7	1	0	2	Xmit	collision
8	2	0	3	Xmit	success
9	1	1	3	Wait	success
10	0	-	-	-	new interval

3.2 Splitting Based on Random Set Selection

Instead of using the source address to decide if we are in the left or right set, flip a coin at each source.

Heads move to the right set

tails move to the left set

Disadvantage:
The number of slots to resolve a collision is unbounded

Advantage
New sources can be added without changing addressing
Fairness: Sources with low addresses do not have a lower delay than sources with high addresses
We can operate with a very large address space, with only a very small subset of the sources active. For instance, the stations may use Internet addresses, but only a small fraction of the sources in the Internet are contending in the local area network.

3.3 Reducing the number of slots needed to resolve contention

3.3.1 Starting Point

— When a collision resolution period (CRP) lasts a long time, there are likely to be many arrivals during the interval.

For instance, when there are m sources, each with a negative exponential distribution between arrivals (when the source is waiting for an arrival), with an arrival rate of λ/m arrivals per slot, and the previous CRP lasts K slots,
the probability that a source has a packet to send when the previous CRP lasts K slots is $1 - e^{-K\lambda/m}$,
the average number of packets waiting to xmit from the m sources is $\bar{s} = m(1 - e^{-K\lambda/m})$.

— If there is more than 1 source waiting to transmit at the beginning of a CRP, there will be a collision.

— If there are $K \gg 2$ sources waiting, and we divide the sources into 2 subsets, it is likely that the transmitting set will also have more than 1 source and that another collision will occur.

We will continue to have collisions and will not have a successful transmission until the number of sources in the transmitting subset is 1.

We can reduce the number of collisions at the beginning of the CRP by initially dividing the sources into more subsets with a smaller average number of sources per subset.

If we divide the sources into too many subsets, the subsets will be empty and the slots will be wasted.

— Intuitively, we should make the initial partitions small enough that there is an average of about 1 source per subset.

Capatenakis showed that the optimum average number of sources per subset is 1.266.

The throughput is about .43 packets per slot, rather than .368 for slotted Aloha, about a 17% improvement

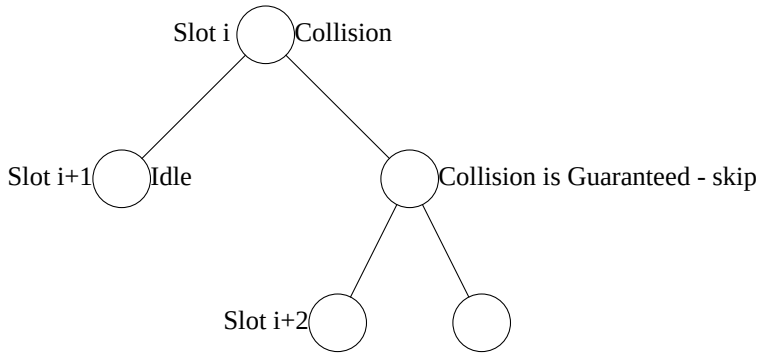
This calculation assumes a Poisson arrival process. The improvement will be much greater when the arrivals are correlated, such as in sensor networks.

3.3.2 Some patterns of collisions and idle slots make it certain that collisions will occur

If a collision is certain in a subset we can skip transmitting the slot and partition the subset immediately into smaller subsets.

When a collision is followed by an idle on the left hand subset, all of the sources that collided must be in

the right hand subset and will collide, We can skip this slot and subdivide the packets immediately



This increases the throughput to about .46, about a 25% improvement over slotted Aloha

3.3.3 When there are multiple sources in the left-hand subset

Indicated by a collision

If the inital slot was selected so that the expected number of sources contending in the tree is 1.266,
and there are 2 or more sources in the left hand branch,
then the expected number of sources in the right hand branch is less then 1.266

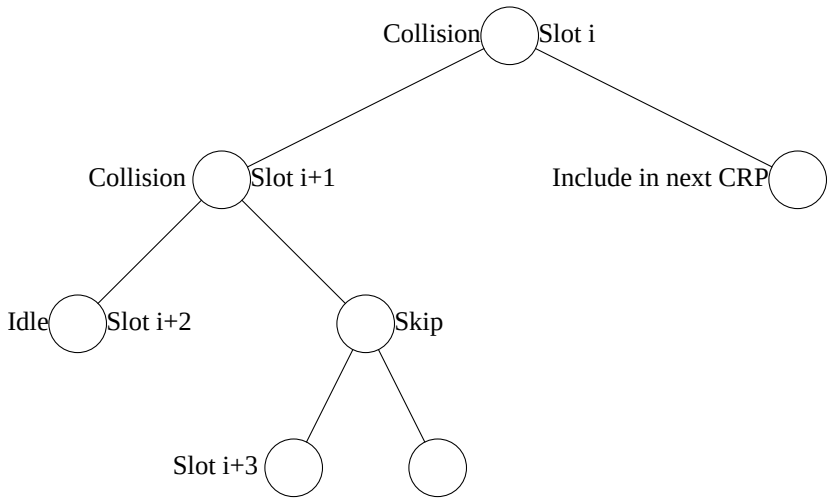
After we resolve the collisions on the left hand side, the right hand subset is smaller than the optimum size subset

Instead of determining if there are any sources in this subset, combine the sources in this subset with the sources waiting for the CRP to end, and pick optimum size subsets.

Note: If there is no transmission or a successful transmission in the left-hand set, then there is at least one source in the right hand set, and we resolve collisions between sources in that set, rather than combining them with more sources.

Gallager shows that the maximum throughput with this scheme is .487, a 32% improvement over slotted Aloha

The increase in capacity is particularly important in satellite networks.



3.4 Splitting Based on Time of Arrival

Instead of randomly splitting sources into sets, we split them depending on the time that the packets arrived, and allow those that arrived first to transmit first.

- Each source keeps track of

  i. the current time *time*, and

  ii. the arrival time of its packet, $x(t)$.

- All sources with a packet that arrived during a contention interval, starting at T_s and ending at T_f, transmit in a slot. Where,

  — All packets that arrived before T_s have been successfully transmitted,

  — and $T_f \leq time$.

- We define $\alpha = T_f - T_s$

- When there are no collisions, T_s is the beginning of the preceding slot, T_f is the end of the preceding slot, and the protocol operates like slotted Aloha. All packets that arrive during a slot transmit in the next slot.

- When a collision occurs we split the contention interval in half.
  The contention interval is from T_s to $T_s + \alpha$
  We place those sources that arrived between T_s and $T_s + \alpha/2$ in the left-hand set,
  and those sources that arrive between $T_s + \alpha/2$ and $T + \alpha$ in the right hand set.
  We resolve collisions in the left set first to guarantee that the packets that arrive first are successfully transmitted first.

We will develop an algorithm to perform splitting in steps.
The first algorithm will be the simplest version of the algorithm.
Successive steps will take into account the three techniques that are used to reduce the number of slots needed to resolve collisions.

We implement the algorithms using a push-down, pop-up stack of the contention intervals. The stack is maintained at each source.
The stack entries are 3-tuples: (T_s, α, Set)
where *Set* is *Left* or *Right*.

3.4.1 **Algorithm 1:** Simple Splitting Algorithm

Objective: To resolve all collisions from time T_s to *time*.

- $T_s = T_f$ The start of this interval is the end of the last interval examined
- $T_f = time$
- $\alpha = T_f - T_s$
- $Push(T_s, \alpha, Right)$

While(stack not empty) {

$Next(T_s, \alpha, Set) = Pop(stack)$

$T_s = Next(T_s)$

$\alpha = Next(\alpha)$

$T_f = T_s + \alpha$

if($T_s \leq x(t) < T_f$)
transmit packet in the slot

if(collision) { /* split interval into 2 intervals
$Push(T_s + \alpha/2, \alpha/2, Right)$
$Push(T_s, \alpha/2, Left)$ }

}

3.4.2 **Algorithm 2:** Avoid Unnecessary Collisions

When an interval is split into a left and right interval, there was a collision.

— The left interval is tried next.

— If the left interval is idle, there will be a collision in the right interval.

— Split the right interval instead of wasting the slot.

- $T_s = T_f$ The start of this interval is the end of the last interval examined
- $T_f = time$
- $\alpha = T_f - T_s$
- $Push(T_s, \alpha, Right)$

While(stack not empty) {

$Next(T_s, \alpha, Set) = Pop(stack)$

$T_s = Next(T_s)$

$\alpha = Next(\alpha)$

$T_f = T_s + \alpha$

if($T_s \leq x(t) < T_f$)
transmit packet in the slot

if(collision) { /* split interval into 2 intervals

$Push(T_s + \alpha/2, \alpha/2, Right)$

$Push(T_s, \alpha/2, Left)$

}

if(idle && $Set = Left$) { /* split right interval into 2 intervals **New**

$Next(T_s, \alpha, Set) = Pop(stack)$

$T_s = Next(T_s)$

$\alpha = Next(\alpha)$

$Push(T_s + \alpha/2, \alpha/2, Right)$

$Push(T_s, \alpha/2, Left)$

}

}

3.4.3 **Algorithm 3:** Better Starting Point

— An average of λ packets arrive per second

— If $\lambda(time - T_s) \gg 1$, there are likely to be many packets in the CRP and slots will be wasted until the contention interval is small enough that there is only one packet in the interval.

— Instead of resolving collisions in the entire time at once, break it into smaller pieces, $T_s \rightarrow T_{f1}$, $T_{f1} \rightarrow T_{f2}$, $\cdots$, $T_{fi} \rightarrow time$ that are resolved during successive CRP's. The final time of one CRP is the starting time for the next.
time keeps moving forward as we resolve the CRP's.

— In an interval $\alpha_0 = T_f - T_s$, there are an average of $\lambda\alpha_0$ packets. If the interval is too small, it is likely that the CRP doesn't have any packet arrivals and an idle slot is wasted.

— There is an optimum value of $\alpha_0 = \alpha_{opt}$ that has the greatest throughput per slot. $\lambda\alpha_{opt}$ is slightly greater than 1.
Bertsekas and Gallagher prove that $\lambda\alpha_{opt} = 1.266$.

— Algorithm 3 is the same as Algorithm 2 except the initial $T_f = \min(time, T_s + \alpha_{opt})$

3.4.4 Algorithm 4: Increase Probability of Success for Multiple Collisions

A set is divided into a left hand and right hand set by a collision.
When there are 2 or more entries in the left hand set, indicated by a collision, we return the right hand set to the next CRP, and examine that set as part of an interval with α_{opt}

- $T_s = T_f$ The start of this interval is the end of the last interval examined
- $T_f = \min(time, T_s + \alpha_{opt})$
- $\alpha = T_f - T_s$
- $Push(T_s, \alpha, Right)$

While(stack not empty) {

$Next(T_s, \alpha, Set) = Pop(stack)$

$T_s = Next(T_s)$

$\alpha = Next(\alpha)$

$T_f = T_s + \alpha$

if($T_s \le x(t) < T_f$)
transmit packet in the slot

if(collision) { /* split interval into 2 intervals

if(Set = Left) Pop(stack) /* remove right half of entry **New**

$Push(T_s + \alpha/2, \alpha/2, Right)$

$Push(T_s, \alpha/2, Left)$

}

if(idle && $Set = Left$) { /* split right interval into 2 intervals

$Next(T_s, \alpha, Set) = Pop(stack)$

$T_s = Next(T_s)$

$\alpha = Next(\alpha)$

$Push(T_s + \alpha/2, \alpha/2, Right)$

$Push(T_s, \alpha/2, Left)$

}

}

At $\lambda = .487$, $\alpha_0 = 1.266/\lambda = 2.6$ slots
At the optimum value our CRP resolves collisions between packets that arrive in a 2.6 slot interval

Go over and understand example figure 4.11, pg 294 of reference [1].

Home work

1. Contention resolution by tree splitting based on addresses

   — There are 16 terminals with binary addresses 0000 to 1111.

   — At the beginning of a contention resolution period, terminals 4,5,8, and 13 have packets to transmit.

   A. Draw the contention resolution tree, showing which terminals can transmit in which slots. (Use the simplest version of the algorithm that does not try to save slots)

      — Number the slots

      — Indicate which slots have collisions, C, which slots are empty, E, and which slots have a successful transmission, S.

      — When there is a collision, indicate which terminals collide in a slot

      — When there is a successful transmission, indicate which terminal succeeds.

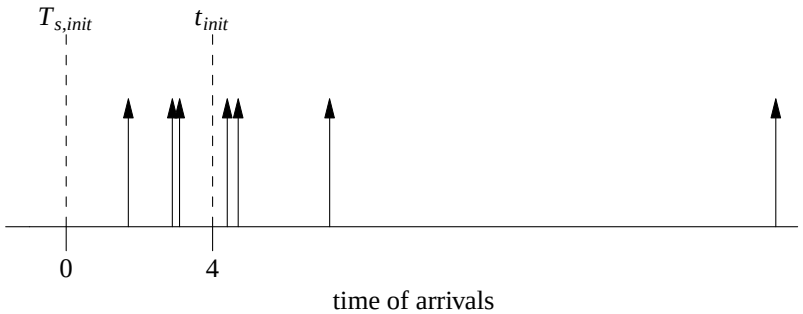
      The contending terminals are 0100, 0101, 1000 and 1101

   B. Which slots could we have avoided transmitting because we knew that there would be a collision or empty slot? Explain how we knew.

   C. Give the values of the counters C_B, C_T and N_B for terminal 5, at the beginning of each slot (before the collision, success, or empty slot has occurred).

2. FCFS splitting

   - Each slot that is transmitted takes 1 unit of time
   - The optimum initial interval is $\alpha_{opt} = 3$ slot times
   - Start the CRP at time $t = 4$. At $t = 4$:
     All packets that arrived before time $t = 0$ have been successfully transmitted
     This CRP follows another CRP, rather than being initiated by a collision.
     Start with $T_S = 0$, $\alpha = \alpha_{opt}$, and a single right hand interval in the stack.
   - Arrivals occur at times 1.7, 2.9, 3.1, 4.4, 4.7, 7.2, and 19.4



time of arrivals

   A. What is the arrival rate λ in arrivals per slot?

   B. Run the CRP from $t = 4$ until the arrival at 19.4 is successfully transmitted.
   Assume that the sequence of CRP's does not terminate.
   Report the results in a table with the following columns:

a. The time at the beginning of the slot that is transmitted in the CRP

b. T_s, the beginning of the time interval that is being resolved

c. α, the size of the interval being resolved.

d. T_f, the end of the interval being resolved

e. The set being resolved:
L = Left-hand, R = Right-hand

f. The state of the slot transmitted on the channel:
Idle, Successful transmission, or Collision
When there is a successful transmission, indicate the arrival time of the packet that succeeds

g. The operation followed by the protocol when the slot is transmitted:
SI = split the interval being resolved,
SL & RR = split a left interval and remove the right part of that interval from the stack.
MF = Move forward to the next interval in the current CRP
MF & SR = Move forward to the next interval and split that Right interval in half.
N_CRP = Start next CRP in the sequence.
Other -- Specify any other operations

4. Contention Resolution in Hardware Systems

Assumptions:

1. A small network:
   All participants receive a bit before the next bit is transmitted.
2. Or'ed Channels:

   Xmit 1's or 0's

   If a 1 and a 0 is transmitted, a 1 is received

   Most fiber-optic networks are baseband
   To send a "1" light is transmitted, to send a "0" no light is transmitted
   In WDM light or no light is transmitted on a wavelength

   Many busses in digital systems and computer systems are baseband
   To send a "1", a voltage is applied to the bus, to send a "0", no voltage is applied

4.1 Simple Polling

Reference [1] section 4.5.6

The system alternates between a contention interval and a data transmission interval.

During the contention interval:

A. There is 1 bit assigned to each possible source
B. If a source has data to send it sends a "1" during its assigned bit, otherwise it doesn't transmit, and the bit is "0".
C. If there are "m" possible sources, and "n" are active, an average of m/n bits of overhead is assigned to each message to resolve contention
   If all "m" sources have data to send, 1 bit per message is used to resolve the contention.
   If 1 source has data to send, then m bits of overhead are assigned to this message

During the data transmission interval:

A. Each source transmits its data packet in the order its "1" appeared during the contention period.
B. If a source's "1" was the i^{th} "1" in the contention period, it waits until $i-1$ other sources transmit before transmitting its data.
C. It isn't necessary for a source to transmit it's address, because the receiver can determine the source address from the order of the 1's during the contention period.

4.2 Simple Splitting

Bit-by-bit Contention Resolution based on Terminal Addresses
This technique has been used in digital switches built by AT&T

In the earlier splitting rule a source's address was used to split sources into groups after a collision occurs, and only the source's in one of the groups contends in the next round. Now, since a source can receive the bits transmitted by all other sources before it transmits its next bit, and because 1's dominate over 0's on the ore'ed channel, sources can split the group of sources as each bit is transmitted and arrange for one source to win every collision.

Operation:

- Sources xmit the bits of their address
- If a 1 is received when a source xmits a 0, it stops transmitting
  Sources defer to sources with higher addresses.
- At the end of the address transmission only one source is transmitting
- That source xmits its data

Problem: If all 0's are transmitted during the address transmission does that mean that a) Terminal $000\cdots0$ has data to transmit or b) No terminal has data to transmit?

This problem can be solved by not using the all "0" address or by using the first bit to poll all stations and using the contention resolution only when there is a station with data to transmit. In the reference [1], Bertsekas and Gallager use the second technique. We will use the first.

Example

- 4 bit binary addresses
- Contending stations 1011, 1000, 0011, 0010

Bit	Xmit	Contending	Waiting	Action
4	1	1011, 1000, 0011, 0010		0011, 0010 Wait
3	0	1011, 1000	0011, 0010	
2	1	1011, 1000	0011, 0010	1000 Waits
1	1	1011	1000, 0011, 0010	1011 Xmits
4	1	1000, 0011, 0010		0011, 0010 Wait
3	0	1000	0011, 0010	
2	0	1000	0011, 0010	
1	0	1000	0011, 0010	1000 Xmits
4	0	0011, 0010		
3	0	0011, 0010		
2	1	0011, 0010		
1	1	0011, 0010		0011 Xmits
4	0	0010		
3	0	0010		
2	1	0010		
1	0	0010		0010 Xmits
4	0			
3	0			
2	0			
1	0			No Xmission End of CRP

- 20 contention bits are xmitted.
- Simple Polling would resolve the contention with 15 bits

  terminal 15->1 , xmit a 1 in their assigned bits if they are contending
  0000,1001,0000,110 $\rightarrow$ terminals 11,8,3,2 want to xmit

- The number of bits in simple splitting can be reduced by realizing that bits 4 and 3 in the 4^{th} address contention and bits 4, 3, and 2 in the 5^{th} must be "0" and did not have to be transmitted.

Efficiency:

- There are $m = 2^k - 1$ possible sources.
- $k = \log_2 m$ bits are transmitted before each data transmission
- If n of the m sources are waiting to transmit, $(n+1)\log_2 m$ bits are transmitted to resolve the contentions.
  The address of each of the sources, plus the all 0's address that indicates the end of the list of contending sources.
- Simple polling transmits m bits to resolve contention.
- Simple polling is more efficient than simple splitting when $m \leq \overline{(n+1)}\log_2 m$.
- Note: The $\bar{n}$ depends on the arrival rate λ of new messages, the average message lenght $\frac{1}{\mu}$ and the duration of the preceeding collision resolution interval. Since, new sources wait for the sources in the preceding CRP to resolve their collisions and transmit their data before contending for the channel.

4.3 Radix R Polling and Splitting Algoritms

The objective is to use polling to successively split the set of sources into smaller and smaller groups.
Instead of limiting ourselves to halving the groups we will consider different bases.

Operation:

Source addresses are expressed base R,
R is the radix
Example: In a radix 3 system, source 23 has address 212 base 3

R bits are used to successively poll the integers of the source addresses

1. Contending sources transmit a 1 in bit positions $R-1,\ R-2, \cdots, 0$, depending on the high order integer of their address.

   Example: Source 212 base 3 transmits a 1 in the first of 3 bits, during the first polling interval.

   All contending sources with addresses 2XX transmit a 1 in this bit position.

   Therefore, 9 sources may transmit a 1 during this bit position.

2. The high order address bits of sources that transmit are placed in a stack

   For instance, if 101 is received during the first contention interval,

   There are sources with address bits 2XX, and 0XX contending.

   0XX, and 2XX are placed on the stack.

3. The stack entries are removed one at a time, and the same procedure is used to poll the next highest order bit of their address - splitting.

   For instance, When 2XX is removed from the stack,

   Contending sources with address bits 22X transmit in the first bit position of the next poll, and so on.

4. When contentions are resolved, the sources transmit their data.

Example

Base 10 addresses

For each position xmit a 1 in polling bits 9,8,...,0

Contending sources 122, 125, 705, 722, 725

Resolve	Position	Xmit	Stack	Action
X-- 7X- 72X	100's 10's 1's	0010000010 0000000101 0000100100	Empty 1X- 70X 1X-	Add 1X- 7X- to stack Add 70X 72X to stack 725 then 722 Xmits
70X	1's	0000100000	1X-	705 Xmits
1X- 12X	10's 1's	0000000100 0000100100	Empty Empty	Add 12x to stack 125 then 122 Xmit

60 bits are required for contention resolution

1000 bits would be required for simple polling

Notes

1. When R= the number of sources, the system becomes simple polling.

   There are 27 sources that can be uniquely identified by 3 integers in a radix 3 system.

   If we used a radix 27 system, instead of a radix 3 system

   A single integer would be used to identify the sources

2. In a radix R system, the R bits in a poll can be transmitted on R separate wires.

   We don't start the next poll until all R bits are transmitted.

   In a backplane we can have R parallel wires for polling and data transmisssion.

   We can construct a radix 8 system that transmits bytes in parallel

Home work

1. Base 5 addresses
   Contending stations 3001, 1444, 1443, 300, 4, 3, 2, 0
   a. Resolve the contention
   b. How many bits are required?
   c. How many bits would be required for simple polling?
2. Polling/Splitting on an Or'ed channel
   The terminals have 10 bit addresses.
   At the beginning of a contention period, *K* stations are waiting to transmit
   A. How many bits are needed to resolve the contention with simple polling?
   B. How many bits are needed to resolve the contention with splitting by bit-by-bit contention resolution based on the terminal address, when address splitting is preceded by an active bit? (Any station that can transmit sends a "1" in the active bit.)
   C. For what values of K does splitting require fewer bits than polling?
   D. In hybrid polling/splitting the high order *j* bits of the address are polled.
   A "1" in the polling bit is the first active bit for the terminals in that group.
   How many bits are needed to resolve the contention?

E. When $j = 5$, for what values of K does hybrid polling/splitting require fewer bits than polling or splitting?

F. What bits are transmitted in hybrid polling/splitting when $j = 3$ and stations (101 0101110), (101 1000010), (101 0101101), (000 0000000), (110 1101001) are waiting to transmit?

5. Carrier Sense Multiple Access

In many networks, such as local area networks, the propagation delay between sources is much less than the time it takes to transmit a packet. In an unslotted, random access system it makes sense to listen to the channel to avoid transmitting on a busy channel. Under certain channel conditions it is also possible to detect collisions while transmitting and to abort transmissions that are destined to fail.

5.1 CSMA - Carrier Sense Multiple Access

Reference [1] section 4.4, except 4.4.1

Transmission Rule:

Listen to the channel before transmitting.
If the channel is busy, don't transmit.

Result:

The source that was transmitting successfully completes its transmission
The channel utilization improves because less time is wasted by colliding transmissions.

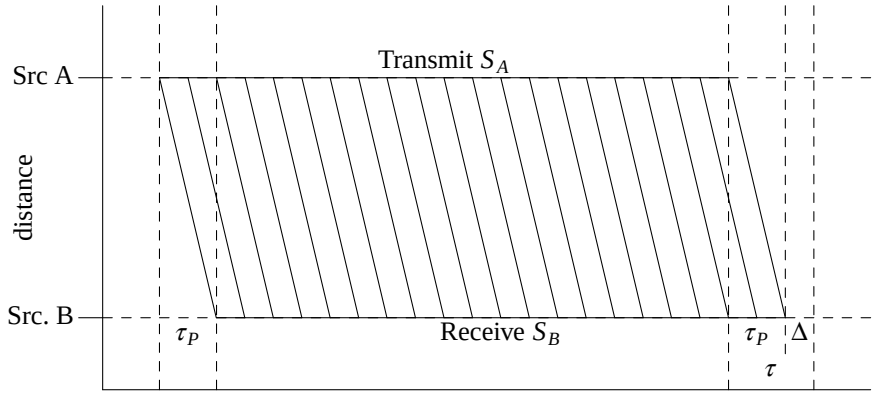
The source that would have collided becomes backlogged and retries later.
Instead of 2 sources becoming backlogged, only 1 source becomes backlogged.

The increase in throughput obtained by CSMA over unslotted Aloha depends upon the ratio of the transmission time and the propagation delay between sources.

On channels with long propagation delays:
When we transmit on a channel that is idle, our signal may still collide at the receiver.
Undetected collisions become more likely as the propagation delay increases.

A signal that is detected at the source may not collide with the sources transmission at its intended receiver.
Unnecessary transmission delays are more likely when the propagation delay is long.

Increasing the transmission rate has the same effect as increasing the propagation delay.
The time that it takes the transmit a packet decreases.
The fraction of the time that a detected transmission provides useful, rather than misleading information about reception at a distant receiver decreases.



τ_p = propagation delay between xmitting source A and detecting source B

Δ = time to detect the absence or presense of signal

$\tau = \tau_p + \Delta$

$T_X = \frac{1}{\mu C}$ = avg packet xmission time

$\beta = \frac{\tau_{\max}}{T_X}$ = time to detect an idle channel or busy channel, normalized wrt the packet xmission time, the fraction of a packet

The packet size on an Ethernet is between 64 and 1518 bytes
512 and 12,144 bits

The transmission rates are 10 Mbps, 100 Mbps and 1000 Mbps

The distances between repeaters is between 100 and 500 meters, depending on the type of cable that is used
A 10Base2 Eternet transmits at 10 Mbps, baseband, with a span up to 200 meters between repeaters
A 10Base5 Eternet transmits at 10 Mbps, baseband, with a span up to 500 meters between repeaters
There may be several spans in series

The speed of light is about 1 ft/nanosec.

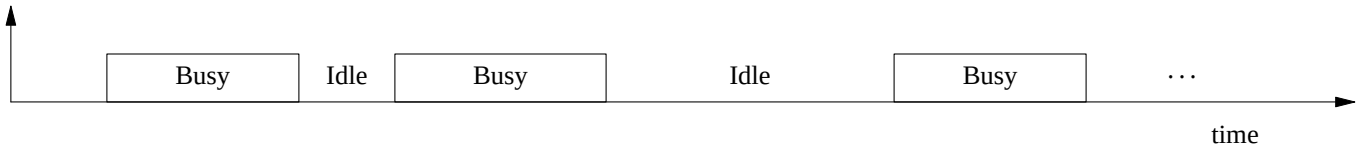
On a 10 Mbps Ethernet:
A bit is 100 ft. long
$\tau_{\max}$ is approximately 30 bits long on a 3000 ft. network, ignoring Δ.
If T_x is 12,000 bits, $\beta = .0025$
If T_x is 512 bits, $\beta = .059$

On a 100 Mbps Eternet:
A bit is 10 ft long
$\tau_{\max}$ is approximately 300 bits long on a 3000 ft. network
If T_x is 12,000 bits, $\beta = .025$
If T_x is 512 bits, $\beta = .59$

5.1.1 An Approximate Analysis of the Throughput.

The exact analysis depends on the relative positions of all of the sources and receivers.
In this analysis we assume that $\beta \ll 1$
We combine components from the analysis in Section 4.4.2 in Bertsekas and Gallager and the analysis in Kleinrock and Tobagi [4]

The transmission pattern on the network consists of alternating busy and idle intervals



The average cycle is $\bar{B} + \bar{I}$, where $\bar{B}$ and $\bar{I}$ are normalized wrt to T_X, the average transmission time.

The probability of successful transmission, no collision, is P_S.

When there is no collision in a cycle an average of 1 unit of T_x get through the network
When there is a collision 0 units get through during the cycle.
Therefore, the throughput per cycle is P_S.

The throughput per transmission time is:

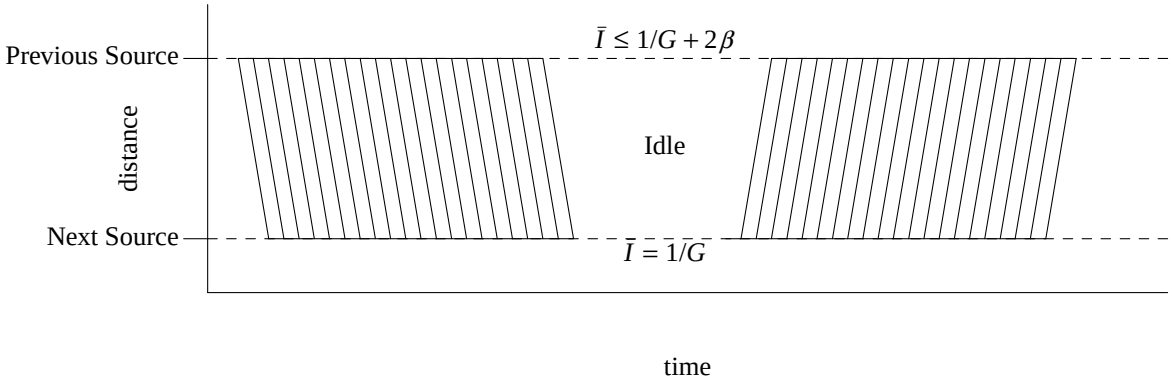
$$S = \frac{P_S}{\bar{I} + \bar{B}}$$

1. Calculate $\bar{I}$

The average arrival rate is G transmissions per T_X.
In our previous, detailed analysis of slotted Aloha systems, G is a function of the number of backlogged sources, n.
In adaptive, slotted systems we then adjusted the retry rate so that G is approximately equal to the value of G that provides the maximum thoughput for all n.
In this analysis, we assume the retry rate is adjusted so that G is approximately constant for all backlogs, n.

$\bar{I}$ at the next source, the average time between detecting the end of transmission and the beginning of its transmission is $\frac{1}{G}$ transmission times.

The lenght of the idle interval that is measured at the different source depends on their position relative to the previous and next source.



$$\frac{1}{G} \le \bar{I} \le \frac{1}{G} + 2\beta$$

In this analysis, we will set $\bar{I} = \frac{1}{G} + \beta$.

2. Calculate P_S

   The first transmission during an idle interval occurs at time t_1

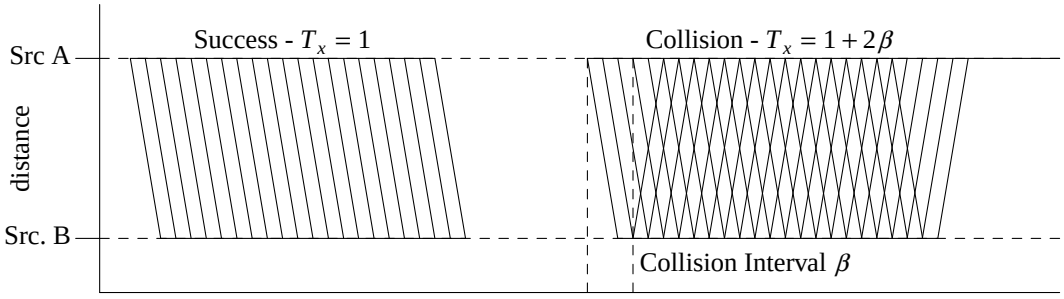
   By $t_1 + \beta$, all other sources detect this packet and do not transmit.

   The transmission is successful if no other messages arrive between t_1 and $t_1 + \beta$.

   $P_S \ge e^{-\beta G}$.

3. Calculate $\bar{B}$

   Successful and unsuccesful packets have different lengths



   Successful packets are 1 message long

   Unsuccessful packets are at most $1 + 2\beta$ messages long

$$\begin{aligned} 1 \le \bar{B} &\le 1 \times P_S + (1 - P_S)(1 + 2\beta) = 1 + 2\beta P_S \\ &\le 1 + 2\beta(1 - e^{-\beta G}) \approx 1 + 2\beta^2 G \approx 1 \end{aligned}$$

   Therefore, $\bar{B} \approx 1$

4. Calculate $S = \frac{P_S}{\bar{B} + \bar{I}} = \frac{e^{-\beta G}}{1/G + \beta + 1} = \frac{G e^{-\beta G}}{1 + G(\beta + 1)}$

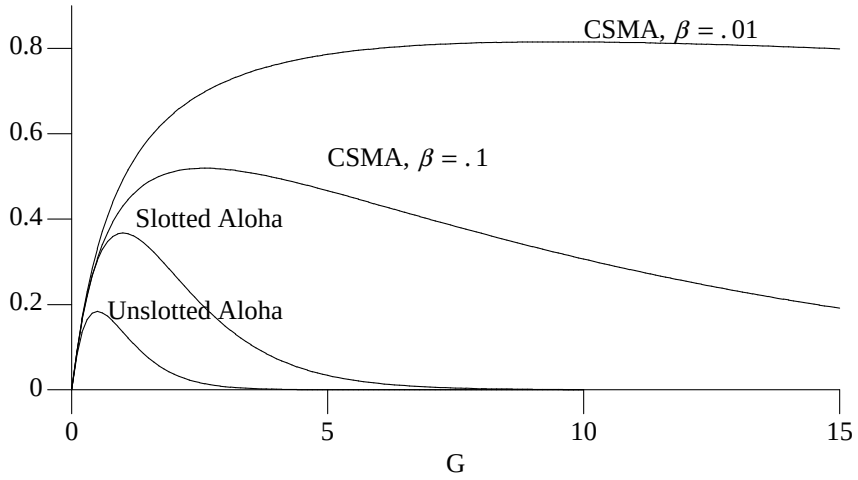
5. To find $S_{\max}$

   Take $\frac{dS}{dG} = 0$

   $G_{opt} \approx \frac{1}{\sqrt{\beta}}$, for $\beta \ll 1$

   $S_{\max} \approx \frac{1}{1 + 2\sqrt{\beta}}$ for $\beta G \ll 1$, so that $e^{\beta G} \approx 1 + \beta G$.

Plotting the departure rate of CSMA, and comparing it with Aloha



We note that:

1. The maximum throughput is significantly higher for CSMA than Aloha
2. The maximum throughput increases as β decreases
3. The curve is still bistable, but we would operate the system in the lightly loaded mode, with 1 stable operating point
4. The curve for CSMA is much flatter than the curves for Aloha
   Adapting G isn't as critical to keep the system near the optimum operating point.
5. Using the approximate analysis for the optimum operating point:

   For β = .1, S_{max} = . 613, and $G_{max} \approx$ 3. 16

   For β = .01, S_{max} = . 83, and $G_{max} \approx$ 10

   The approximation is reasonably good for β=.01, but not β=.1

5.2 CSMA/CD

Reference [1] section 4.5.2

Transmission Rule:

Listen to the channel before xmitting as in CSMA

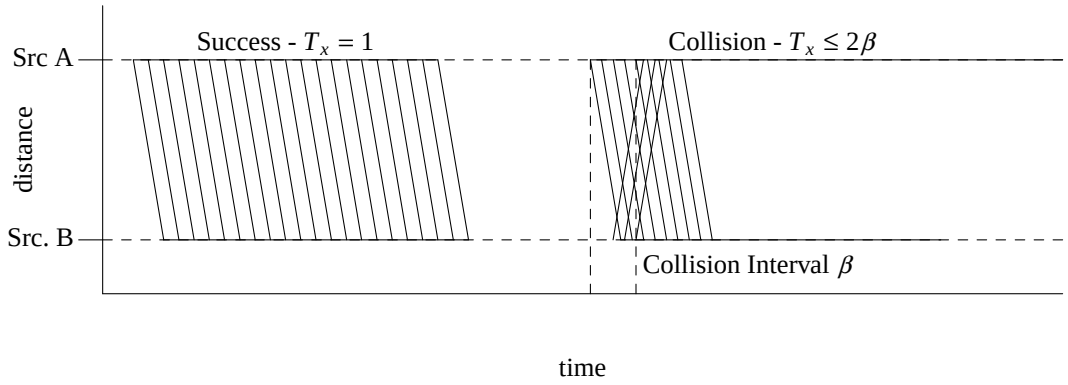
Listen while Xmitting, and stop if a collision is detected

Implementation:

Simple on wired LANS - terminated lines -> no reflections

In radio networks reflections of our own signal are confused with other terminals

The maximum length of a colliding packet is 2B



5.2.1 Analysis

$\bar{I} < \frac{1}{G} + \beta$ - As in CSMA

$P_S > e^{-\beta G}$ - As in CSMA

$\bar{B} < 1 \times P_S + (1 - P_S)(2\beta) = 1 - (1 - 2\beta)(1 - P_s)$

Unsuccessful packets have a maximum length of 2β instead of $1 + 2\beta$

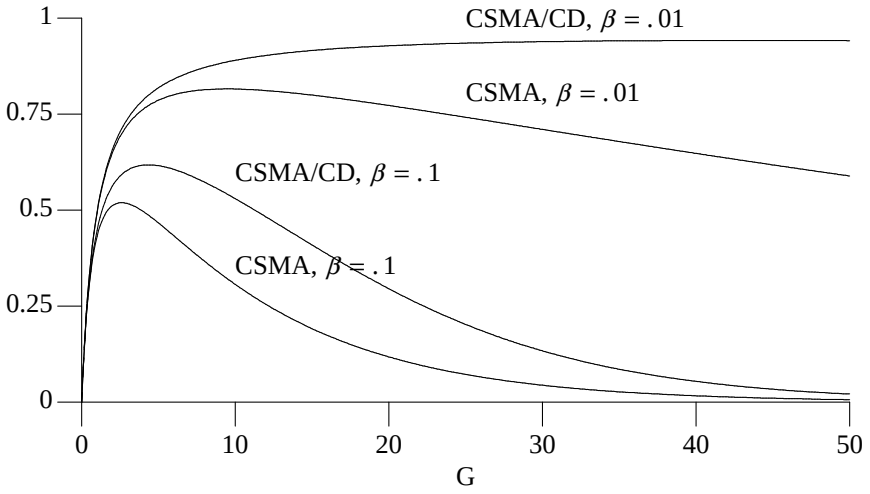
$\bar{B} < 1 - (1 - 2\beta)(1 - e^{-\beta G})$

$$S = \frac{P_S}{\bar{B} + \bar{I}} > \frac{e^{-\beta G}}{1/G + \beta + 1 - (1 - 2\beta)(1 - e^{-\beta G})} = \frac{Ge^{-\beta G}}{1 + G(\beta + 1) - G(1 - 2\beta)(1 - e^{-\beta G})}$$

$$S_{CSMA/CD} = \frac{1 + G(\beta + 1)}{1 + G(\beta + 1) - G(1 - 2\beta)(1 - e^{-\beta G})} S_{CSMA}$$

$S_{CSMA/CD} > S_{CSMA}$ for all β, G

Plotting the throughput for CSMA and CSMA/CD



According to Gallager:

$$S_{\max} \approx \frac{1}{1+6.2\beta}, G \approx \frac{.43}{\beta}$$

For $\beta = .1$

$S_{\max} = .617, G_{\max} = 4.3$

This approximation is closer than that for CSMA

For $\beta = .01$

$S_{\max} = .94, G_{\max} = 43$

This is the value of β in 10 Mbps Ethernets

The maximum throughput is almost 95% of that obtained by perfect scheduling

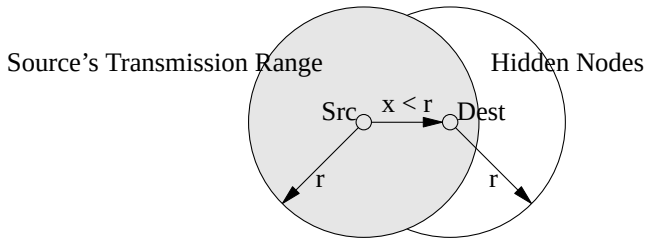
5.3 CSMA with Collision Avoidance (CSMA/CA)

Reference [5], section 4.4.

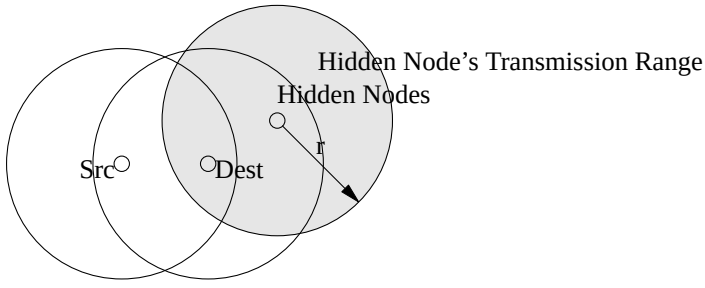
5.3.1 Problems: Hidden Nodes and Exposed Nodes

5.3.1.1 Hidden Nodes

A. **Problems in CSMA**



a. The hidden node can't hear the source transmit,
but if the source is transmitting,
the hidden node interferes with the source at the destination.



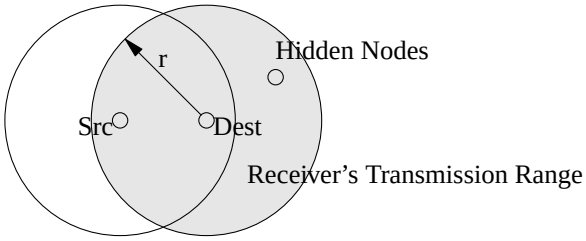
b. The source can't hear the hidden node transmit,
but if the hidden node is transmitting,
the source cannot successfully transmit to the destination

B. **Solutions:**

a. *3 way hand shake: (CSMA/CA) with ACK*

— If the source doesn't detect a busy channel, it sends RTS,

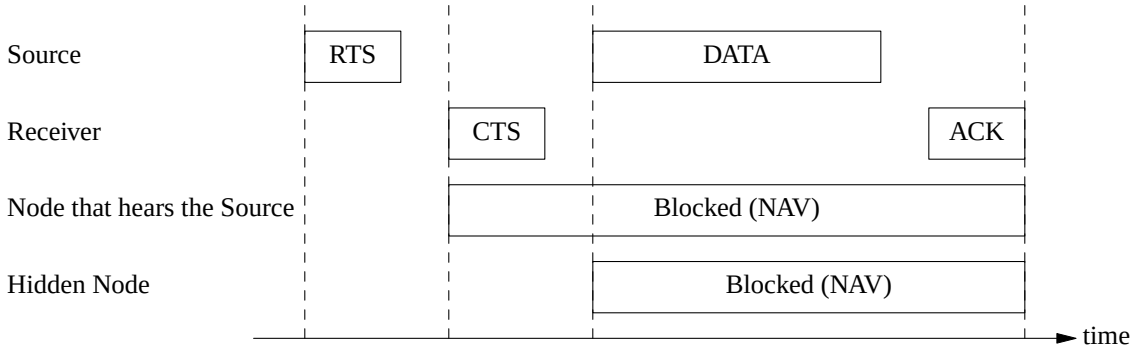
— If a hidden node isn't transmitting, the receiver receives RTS from the source and sends CTS,



— All hidden nodes hear the CTS and consider the channel to be busy for a packet transmission time,

— The source hears the CTS and transmits the data

— ACK's aren't used in wired LAN's because the probability of an error in the transmission is low, however, the probability of a transmission error in wireless networks is much higher than in wired networks



b. *Busy Tone: (In a separate frequency band)*

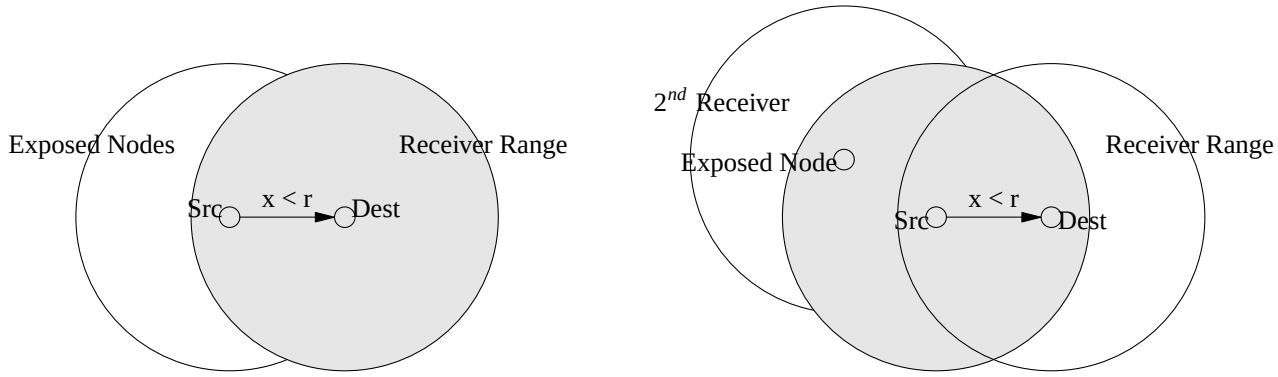
— If the source doesn't detect a busy channel, or hears a busy tone, it sends data,

— If a hidden node isn't transmitting, the receiver sends the busy tone

— All hidden nodes hear the busy tone and don't transmit

— If the source doesn't hear the busy tone it stops transmitting

5.3.1.2 Exposed Nodes



A. **Problem:** Exposed nodes reduce the number of simultaneous transmissions

— Up to 40% of the nodes that are blocked when they hear a source transmit can transmit without interfering with the receiver (the exposed nodes).

— Up to 60% of the nodes that the exposed nodes can reach do not receive the transmission from the transmitting source.

B. Solutions:

a. **3-way Handshake:**
Potential Source is inhibited when it receives CTS from a receiver,
But, not when it receives RTS or Data from a source

b. **Busy Tone:**
Potential Source is inhibited while it receives busy tone from a receiver,
But, not when it detects transmission from a source

5.3.2 The IEEE 802.11 Standard

Wi Fi (Wireless fidelity)

The 802.11 Family

802.11a

Data rates: 1, 2, 6, 9, 12, 18, 24, 36, 48, and 54 Mbps

5.8 Ghz band

802.11b

Data rates: 1, 2, 6, and 12 Mbps

2.4 Ghz band

802.11g

switches between 802.11a and 802.11b

— The different data rates allow devices that are closer, and have a higher SNR to transmit at higher rates.

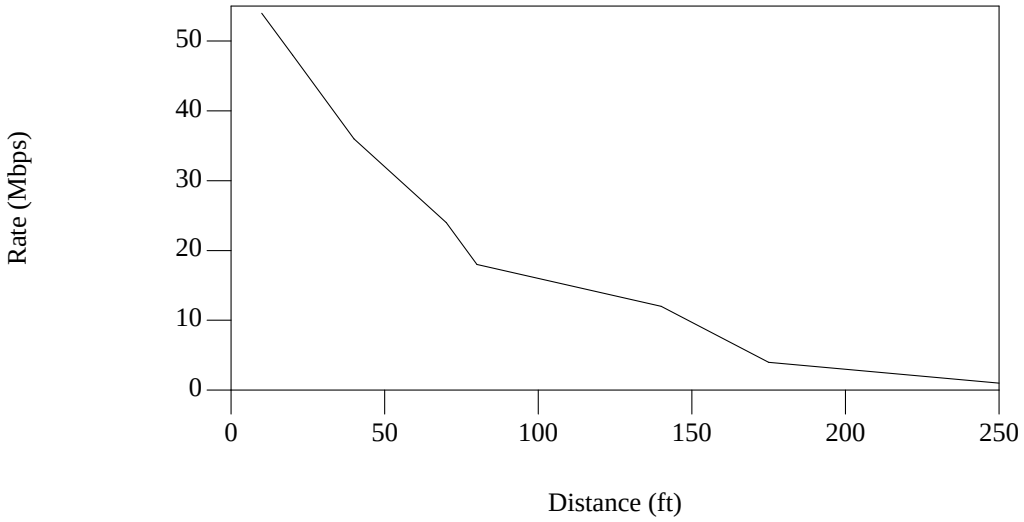
— The same shared channel may have devices transmitting at different rates.

— During the operation of the system, devices may change their rate to increase the probability of successful transmission

Approximate distance(in feet) vrs. transmission rate

	802.11a	802.11b	802.11g
Rate (MBPS)			
1	250	200	250
2	175	175	175
6	175	100	175
9	140	70	140
12	140	40	140
18	80		80
24	70		70
36	40		40
48	20		30
54	10		20

Approx. Rate as a Function of Distance



The rate decreases approximately as $\frac{1}{d^2}$

MAC protocols

1. Distributed Coordination Function (DCF)
   A. Mode 1
      — CSMA
      — Susceptible to collision with hidden nodes
      — Originally it was assumed that all terminals are in the same tranmission range, but IEEE 802.11 is increasingly used in multihop networks.
   B. Mode 2
      — CSMA/CA with Acknowledgement
      — 3-way handshake

- source sends RTS, with packet length
- receiver sends CTS if there is no hidden node transmitting
- other sources are inhibited for message transmission time plus the time to transmit the ACK if they hear the RTS or CTS
  (802.11 does not take care of exposed nodes)

— Ack is used because of high channel error rate

- verifying that a message is correctly received does not require a separate channel contention

C. Multiple packet messages

— The probability of error in the wireless network is usually higher than that in a wired network
Large packets are divided into shorter packets to increase the probability of correct reception, and to retransmit less of the packet when there is an error.

— Successive packets are transmitted when the ACK for the previous packet is received without using the RTS/CTS handshake

— A transmitted packet is the RTS for the next packet, and the ACK is the CTS for this packet.

— The channel is empty for a packet transmission time following the last packet in a message.
This is reasonable when a message consists of many short packets.

— When a transmission error occurs, an ACK is not transmitted, and the source sends an RTS before retransmitting the lost packet.

2. Point Coordination Function (PCF)

— Optional

— A base station transmits a polling sequence to invite sources to transmit

— No collisions

— Periodically, the polling slot invites terminals to send a first frame to join the polling sequence

— There is contention for the invitation, and there may be collisions

— Used for QoS

— The polling order and frequency is not part of the standard and is adapted to the application

3. Combined DCF and PCF

— Time is divided.
Sources shift from polling for a period, followed by DCF for a period.

— This part of the standard is still being developed to allow other management functions.

Home work

1. Consider 4 systems, Aloha, Slotted Aloha, CSMA, and CSMA/CD

   $\beta = .1$

   There is are 100 users.

   Users that are not backlogged or actively, have a Poisson arrival process with $\lambda_A = .01$. With N backlogged users, the arrival rate is $S(N) = (100 - N)\lambda_A$.

   Each system uses random retrys for backlogged users.

The retry process for each backlogged user approximates a Poisson process with λ_r. With N backlogged users, the offered traffic is $G(N) = S(N) + N\lambda_r$

For each of the 4 systems

A. What is the value of G that results in the maximum throughput?

B. What is the maximum throughput?

C. What is the number of backlogged users at the maximum throughput?

D. What value of λ_r must be selected to achieve maximum throughput?

E. What is the average number of retrys?

F. What is the average packet delay?

2. Delay and throughput of ARQ systems with wireless and wired links:

   - An ARQ system periodically retransmits a message until the message is received at the destination and the acknowledgement is received at the source.
   - An acknowledgement for a message is sent from the destination to the source whenever the desstination receives a message, even if the destination has previously acknowledged the message. The assumption is that the earlier ackowledgements were not received by the source.
   - The source handles one message at time. It does not start transmitting the next message until the acknowledgement for the current message is successfully received.
   - The throughput for the system is one over the average interval between the start of transmission of the source message.
   - The average delay, $\bar{X}$, is the average time until the first copy of a message is received by the destination.

   A. General network:

      - The average delay to transmit a message from a source to the destination is $\bar{D}/2$, and the probability that it is lost is $P_{L,M}$.
      - The average delay to transmit an acknowledgement from the destination to the source is $\bar{D}/2$ and the probability that the acknowledgement is lost is $P_{L,A}$.
      - The time that a source waits before retranmsitting a message is 4 times the average round trip delay, 4 * $\bar{D}$.

      a. What is the average number of times that a message is transmitted until it is successfully received by the destination?

      b. What is the average delay, $\bar{X}$?

      c. What is the probability that an acknowledgement is received when a message is transmitted?

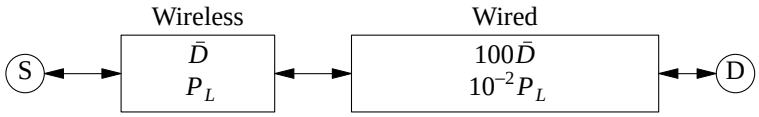
      d. What is the average number of times that a message is transmitted until it the acknowledgement is successfully received by the source?

      e. What is the throughput?

   B. Consider a network with a wireless access local network at the source that connects to a wired network.

      - The average delay on the wireless network is $\bar{D}$ in each direction and the average delay on the wired network is $100\bar{D}$ in each direction.

- The probability of loss on the wireless link is P_L in each direction and the probability of loss in the wired network is $10^{-2}P_L$ in each direction.



For a network with

a. End-to-end acknowledgements between the destination and source, and

   i. What is the approximate round trip delay?

   ii. What is the probability of loss on the forward link, $P_{L,F}$, and the probability of loss on the round trip, $P_{L,RT}$, in terms of $P_{L,F}$?

   iii. The retry interval is 4 times the average round trip delay.
   What is the average time between the start of transmissions from the source, $\bar{X}$, in terms of $P_{L,RT}$?

   iv. Represent $\bar{X}$ as the first 2 terms of the Taylor series expansion in terms of P_L, at $P_L = 0$.

b. End-to-end acknowledgements between the source and destination and also message recovery and acknowledgements on the wireless links.
   The output node on the wireless link sends acknowledgements to the input node on the wireless link and forwards the message as soon as the first copy is received. The input node on the wireless link retransmits the message every $8D_L$, 4 times the average round trip delay in the wireless network, until it receives the acknowledgement.

   i. What is the average delay to forward a message on the wireless link, $\bar{X}_W$?

   ii. What is the average source-destination round trip delay, $\bar{X}$?

   iii. What is the probability of loss on the round trip, $P_{L,RT}$?

   iv. The retry interval is 4 times the average round trip delay.
   What is the average time between the start of transmissions from the source, $\bar{X}$, in terms of $P_{L,RT}$?

   v. Represent $\bar{X}$ as the first 2 terms of the Taylor series expansion in terms of P_L, at $P_L = 0$.

6. Some Other Wireless Standards

6.1 Bluetooth

Architecture

— PicoNet: A master Node and up to 7 active slave nodes within 10 meters

- There can also be up to 255 parked nodes
- A parked node is in a power saving mode waiting to be activated

— Scatternet: An Interconnected collection of PicoNets

- Interconnected by bridge nodes
- A bridge node is a slave in 2 piconets, and interconnects them

MAC

— The master polls the slaves

— All communications is Master to Slave
Never slave-to-slave

— Alternate slots for master and slaves
Master gets 1/2 of the slots

— The total bit rate is 1 MBPS

6.2 802.16 MAN Standard

- Connect buildings (slaves) to a base station
Cellular radio model
- Connection Oriented

  — The base station polls the users

  — If the user doesn’t respond to 7 consecutive polls it is placed in an idle group

  — There is a poll for all idle stations
  Idle stations contend for this poll, and may collide
- The transmission rate of a station depends on its distance from the base station.

  — If the bandwidth allocated is 25 MHz

  — A nearby station transmits at 6 bits/baud and transmits at 150 Mbps during its slot

  — A medium distance station transmits at 4 bits/baud and transmits at 100 Mbps during its slot

  — a long distant station transmits at 2 bits/baud and transmits at 50 Mbps during its slot

  — A user trys a transmission rate, and decreases the rate if the error rate is too high - like 802.11
- Bandwidth splitting

  — The number of slots given to the base station and the slaves varys depending on the application.

  — Internet has more traffic downstream - to the stations - than upstream - to the Internet

  — Therefore, in this application, the master gets more slots to transmit to a slave than the slave gets to transmit to the basestation.

7. MANET Protocols

MAC Protocols used in Multihop Networks [6]

Difference between a LAN and a Multihop Network:
In a multihop network nodes try to make as much progress toward the destination as possible. Most transmissions are close to the max. transmission distance.
There are more hidden and exposed nodes.

7.1 Sender Initiated

7.1.1 MACA - Multiple Acess with Collision Avoidance

- RTS/CTS/Data, 3-way handshake, like IEEE 802.11
  - — RTS-CTS is the contention period. The data transmission period is contention free
- In addition, a source knows how much power it needs to reach each of the receivers.
- Increase throughput by power control:
  - — If A hears CTS from B, and knows how much power it needs to interfere at B, it can still communicate with nodes that require less power.
  - — This is much more complicated than just having exposed nodes try to communicate.

7.1.2 Power-Aware Multiple-Access Protocol (PAMA)

- Conserving power is a concern
- MACA is used with 802.11
- In 802.11, the NAV signal indicates time periods when a souce cannot transmit.
- In PAMA, the transceiver power is turned off during this interval

7.1.3 Busy tone - multiple access BTMA

- When a receiver starts receiving data, it transmits a busy tone to prevent nodes that are hidden from the source from also transmitting
- The busy tone serves the same function as the CTS in 802.11, it notifies hidden nodes when the receiver is busy
- The busy tone from the receiver is more durable than the CTS signal.
  - — The CTS may get lost, either because of a transmission error or because a source that may receive the CTS is receiving a packet from a source that is hidden from the receiver, and not set the NAV
  - — The tone is continuous and will be present when the hidden stops transmitting.

7.1.4 Dual busy tone multiple acess - DBTMA

- Use both RTS/CTS protocol from 802.11, and receiver Busy tone from BTMA
- There is a second tone that is transmitted while the source is sending data.
  - — Other sources detect this tone before accessing the channel
  - — The second tone is redundant with the data transmission, but can be used to inhibit transmission over a larger range than the data reception.
    This can reduce the probability of packet loss at the receiver.

7.2 Receiver Initiated

7.2.1 Polling

Polling is considered a receiver initiated protocol when a single receiver succesively invites sources to transmit.

This is the case in Bluetooth, where there is a master station and each station transmits to, or receives daa from that master station.

Polling is also part of the IEEE 802.11 protocol. In these networks the objective is to connect local sources to a backbone network, and the station that connects with the backbone is the single receiver.

Polling is more general. On a token passing bus, a token can be passed to allow each source to transmit. The source can transmit to any, or all, of the other sources.

7.2.2 MACA-BI (MACA by invitation)

Packets in wireless networks are shorter than those in wired networks because the probability of error is higher

- RTS/CTS for first packet in a long message.
- 1 message - invitation RTR (ready to receive) - instead of 2 messages RTS/CTS for remaining packets in the message
- RTR serves the same function as the ACK in one version of the IEEE 802.11 protocol.

7.2.3 MARCH: Media Access with reduced handshakes

- The first source on a path uses the RTS/CTS protocol to transmit the packet to the first destination.
- If the second destination is not in an active transmission region, it hears the CTS from the first destination to the first source.
- The second destination realizes that the packet arriving at the first destination will be forwarded to it.
- Rather than waiting for the second destination to send an RTS, it sends it a CTS.
- If the second destination is in an active area, the first destination will use the RTS/CTS protocol.
- If the system is lightly utilized, a packet can move from the source, through the intermediate nodes, with only one RTS.

REFERENCES

[1] D. Bertsekas, R. G. Gallager, **Data Networks**, Prentice-Hall Inc, 1992.

[2] N. Abramson, "The Aloha System - Another Alternative for Computer Communications," Fall Joint Computer Conference, AFIPS Conference Proceedings, Vol. 37, pp. 281-285, 1970.

[3] N. Abramson, "The Throughput of Packet Broadcasting Channels," IEEE Trans. on Commun., vol. COM-25, no. 1, Jan. 1977, pp 117-128.

[4] L. Kleinrock, F. A. Tobagi, "Packet Switching in Radio Channels: Part I-Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics," IEEE Trans. on Commun., vol. COM-23, no.12, Dec. 1975, pp 1400-1416.

[5] A. S. Tanenbaum, Computer Networks, 4th edition, Prentice-Hall 2002

[6] C.K. Toh, "Ad Hoc Mobile Wireless Networks," Prentice Hall 2002.